S&H Form: PTO/SB/05 (12/97)

JC644 U.S. PTO
03/20/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 826.1593/JDH

First Named Inventor or Application Identifier:

Ryuichi SUNAYAMA, et al.

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, DC 20231

JC542 U.S. PTO
09/528714
03/20/00

1. [X] Fee Transmittal Form
2. [X] Specification, Claims & Abstract ..... [Total Pages: 47]
3. [X] Drawing(s) *(35 USC 113)* ..... [Total Sheets: 12]
4. [X] Oath or Declaration ..... [Total Pages: 34]
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 17 completed)*
      i. [ ] DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. [ ] Incorporation by Reference (usable if Box 4b is checked)
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. [ ] Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))
9. [ ] 37 CFR 3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [X] Information Disclosure Statement (IDS)/PTO-1449[X] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] Small Entity Statement(s) [ ] Statement filed in prior application, status still proper and desired.
15. [X] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. [ ] Other:

17. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: ___/_____

18. CORRESPONDENCE ADDRESS

STAAS & HALSEY, LLP
Attn: James D. Halsey, Jr.
700 Eleventh Street, N.W., Suite 500
Washington, DC 20001

Telephone: (202) 434-1500
Facsimile: (202) 434-1501

© 1997 Staas & Halsey

APPLICATION FOR

UNITED STATES LETTERS PATENT

SPECIFICATION

Inventor(s): Ryuichi SUNAYAMA and Aiichiro INOUE

Title of the Invention: INSTRUCTION PROCESSING DEVICE AND METHOD FOR CONTROLLING BRANCH INSTRUCTION ACCOMPANIED BY MODE CHANGE

INSTRUCTION PROCESSING DEVICE AND METHOD FOR CONTROLLING BRANCH INSTRUCTION ACCOMPANIED BY MODE CHANGE

Background of the Invention

Field of the Invention

The present invention relates to an information processing apparatus adopting an out-of-order process system. In particular, the present invention relates to an instruction processing device for controlling a branch instruction accompanied by a mode change by way of hardware in an information processing apparatus with a plurality of ports for an instruction fetch, and a method thereof.

Description of the Related Art

In an information processing apparatus adopting an out-of-order processing system, subsequent instruction strings are sequentially inputted to a plurality of pipelines before one instruction execution is completed, and performance is improved by executing the instructions using those pipelines.

However, if the execution result of a present instruction affects the execution of a subsequent instruction, the subsequent instruction cannot be

executed before the execution of the present instruction is completed. If an instruction process which affects the execution of a subsequent instruction is delayed, a subsequent instruction cannot be executed during the delay and continues to wait for the completion of the execution of the present instruction. Accordingly, disorder occurs in a pipeline and the performance of an instruction process is degraded.

A branch instruction is a typical example of such an instruction. In a branch instruction, since whether the branch occurs is unknown until the execution is completed and the target address of the branch destination is also unknown, a subsequent instruction is held and the pipeline is disturbed.

Under these circumstances, the high-speed execution of a branch instruction is promoted by using a branch history (branch prediction table). If a branch prediction using a branch history is made when a branch instruction is executed, a subsequent instruction and a branch destination instruction can be processed before it is determined whether the branch occurs.

According to this process system, if a branch instruction is executed, and as a result, it is found

that the branch occurs, the instruction address of a branch destination and the instruction address of the branch instruction itself are paired and registered in a branch history in advance. When the instruction is extracted from a main storage device, the branch history is retrieved prior to the execution of the instruction.

In this way, pairs of the instruction address of a branch instruction that is previously branched and the instruction address of a branch destination are registered in a branch history. A branch instruction which is not registered in a branch history and has been branched must be newly registered in the branch history. If the target address of the branch destination of a branch instruction is changed for some reason, the target address of the branch destination obtained by retrieving it from the branch history becomes invalid.

Out of the entries of a branch history, entries corresponding to the following cases are considered to have a high possibility of failing in a branch prediction and are removed from the branch history:

(1) a prediction that a branch is taken fails twice in succession

(2) a prediction that a branch is taken fails and the

target address of a branch destination is also incorrect

(3) an instruction is registered by mistake although the instruction is not actually a branch instruction

If a prediction that a branch is taken fails or the target address of a branch destination is incorrect, such information must be stored for each target.

However, the conventional instruction process described above has the following problem.

According to the conventional instruction processing device, since an address mode indicating the bit size of an address space is determined by software, the address mode is sometimes changed while instruction is executed. However, when the branch prediction of a branch instruction accompanied by an address mode change is made, only the instruction address of a branch destination can be predicted since a mechanism for predicting the address mode of a branch destination is not provided. Therefore, the pre-fetch of a branch destination instruction string cannot be performed based on the changed address mode, and thereby a correct instruction process cannot be executed.

When the instruction address of a branch

destination is calculated and obtained using an address generator, the address mode of the branch destination can not also be obtained. When the pre-fetch of an instruction is performed at a specific port in an instruction process device provided with a plurality of instruction fetch ports and then the fetch is attempted to be restored to a sequential instruction fetch at another port, a correct instruction process cannot also be executed if the address modes of the respective ports are different.

In this way and according to the conventional instruction process device, if an address mode is changed, a correct instruction process is sometimes not executed. Accordingly, control using a micro-program becomes necessary and thereby the number of clocks needed for the control increases, which is a problem.

Summary of the Invention

The object of the present invention is to provide an instruction processing device for performing a higher-speed instruction process in an information processing apparatus adopting an out-of-order process system, and a method thereof.

In the first aspect of the present invention, an

instruction process device comprises a storage circuit, a branch instruction control circuit and a transfer circuit. The storage circuit stores the address mode information as well as the instruction address of a fetched instruction. If the fetched instruction is a branch instruction, the branch instruction control circuit controls the branch instruction using the address mode information. When the branch instruction is executed, the transfer circuit transfers the address mode information stored in the storage circuit to the branch instruction control circuit.

In the second aspect of the present invention, an instruction process device comprises a storage circuit, a fetch circuit and a control circuit. The fetch circuit fetches an instruction. The storage circuit stores the address mode information of each fetched instruction as a part of an instruction address. The control circuit controls the instruction process of each instruction based on the stored address mode information.

In the third aspect of the present invention, an instruction process device comprises a storage circuit, a fetch circuit and a plurality of instruction fetch ports, and performs an instruction

fetch by way of an out-of-order system. The storage circuit stores the address mode information obtained when an instruction fetch request is issued, as well as an instruction address for each port. The fetch circuit performs an instruction fetch based on address mode information corresponding to a port to be used.

Brief Descriptions of the Drawings

Fig. 1 shows the basic configuration of an instruction processing device of the present invention.

Fig. 2 shows the detailed configuration of an instruction processing device.

Fig. 3 shows an address mode designation bit.

Fig. 4 shows the format of the entry of a branch history.

Fig. 5 is a flowchart showing the operation of an instruction processing device.

Fig. 6 shows signals used for an instruction process.

Fig. 7 shows an address mode storage circuit.

Fig. 8 shows the first address mode generation circuit.

Fig. 9 shows the second address mode generation circuit.

Fig. 10 shows an address mode designation bit generation circuit.

Fig. 11 shows a judgment circuit.

Fig. 12 shows a control signal generation circuit.

Description of the Preferred Embodiments

The preferred embodiments of the present invention are described in detail with reference to the drawings.

Fig. 1 shows the basic configuration of an instruction processing device of the present invention.

In the first aspect of the present invention, an instruction processing device comprises a storage circuit 1, a branch instruction control circuit 2 and a transfer circuit 3. The storage circuit 1 stores the address mode information of a fetched instruction as well as the instruction address of the fetched instruction. If the fetched instruction is a branch instruction, the branch instruction control circuit 2 controls the branch instruction using the address mode information. When the branch instruction is executed, the transfer circuit 3 transfers the address mode information stored in the storage circuit 1 to

the branch instruction control circuit 2.

The address mode information of the fetched instruction is stored with the instruction address in the storage circuit 1, and is transferred to the branch instruction control circuit 2 by the transfer circuit 3. The branch instruction control circuit 2 controls a branch instruction using the transferred address mode information.

According to such an instruction processing device, the address mode information of a fetched branch instruction is automatically transferred from an instruction fetch pipeline to an instruction execution pipeline. Then, if the branch instruction is accompanied by an address mode change, the branch instruction control circuit 2 can adopt the address mode after change, which is designated by the branch instruction, as the address mode of a branch destination. If the branch instruction is not accompanied by an address mode change, the branch instruction control circuit 2 can adopt the transferred address mode information as the address mode of a branch destination.

Therefore, a correct branch destination address mode is automatically determined. Accordingly, there is no need for control using a micro-program and

thereby an instruction process speed can be improved.

In the second aspect of the present invention, an instruction processing device comprises a storage circuit 1, a fetch circuit 4 and a control circuit 5. The fetch circuit 4 fetches an instruction. The storage circuit 1 stores the address mode information of each fetched instruction as a part of an instruction address. The control circuit 5 controls the instruction process of each instruction based on the stored address mode information.

The address mode information of the instruction fetched by the fetch circuit 4 is stored in the storage circuit 1 as a part of the instruction address, and the control circuit 5 controls an instruction process using the stored address mode information.

According to such an instruction processing device, since the address mode information of each fetched instruction can be stored as a part of an instruction address, there is no need to provide a new register, etc., for exclusive use with address mode information. If the branch instruction is accompanied by an address mode change, the control circuit 5 can adopt the address mode after change, which is designated by the branch instruction, as the address

mode of a branch destination. If the branch instruction is not accompanied by an address mode change, the control circuit 5 can adopt the stored address mode information as the address mode of a branch destination.

Therefore, a correct branch destination address mode is automatically determined. Accordingly, there is no need for control using a micro-program and thereby an instruction process speed can be improved.

In the third aspect of the present invention, an instruction processing device comprises a storage circuit 1, a fetch circuit 4 and a plurality of instruction fetch ports, and performs an instruction fetch by way of an out-of-order system. The storage circuit 1 stores the address mode information obtained when an instruction fetch request is issued, as well as an instruction address for each port. The fetch circuit 4 performs an instruction fetch based on address mode information corresponding to a port to be used.

When an instruction fetch request is issued in an instruction fetch pipeline, the address mode information and the instruction address of the instruction to be fetched are related to a port to be used and stored in the storage circuit 1. The fetch

circuit 4 interprets the instruction address based on the address mode information corresponding to the port to be used, and fetches the instruction.

According to such an instruction processing device, address mode information can be assigned to the respective instruction fetch ports. Therefore, a correct address mode can be automatically determined for each port even if there is a switchover of ports due to the pre-fetch of a branch destination instruction string. Accordingly, there is no need for control using a micro-program and thereby instruction process speed can be improved.

For example, the storage circuit 1 and transfer circuit 3 shown in Fig. 1 correspond to an instruction address register 11, an instruction address generation circuit 12, an instruction fetch control circuit 13, an instruction buffer 16, a decoder 17, a branch destination address generation circuit 19 and the branch instruction control circuit 20 shown in Fig. 2, which is described later, and the branch instruction control circuit 2 shown in Fig. 1 corresponds to a branch instruction control circuit 20 shown in Fig. 2.

For example, the fetch circuit 4 shown in Fig. 1 corresponds to the instruction address register 11,

the instruction address generation circuit 12 and the instruction fetch control circuit 13 shown in Fig. 2, and the control circuit 5 corresponds to an instruction control circuit 18, the branch destination address generation circuit 19, the branch instruction control circuit 20 and a PSW (Program-Status-Word) control circuit 21 shown in Fig. 2.

In this preferred embodiment, since a branch prediction for a branch instruction accompanied by an address mode change is made, the address mode information, as well as instruction address of a branch destination, are stored in a branch history. For example, a bit with the address mode information of a branch destination is stored as a part of the instruction address of the branch destination. Then, in the branch instruction control unit, it is judged whether a branch destination address used when a branch prediction is made, is correct and simultaneously it is judged whether the correct address mode of a branch destination is designated.

A bit with the address mode information of a branch destination is generated by the branch instruction control unit. The branch instruction control unit judges whether a branch instruction is accompanied by an address mode change. If the branch

instruction is accompanied by an address mode change, the branch instruction control unit obtains information about an address mode after change, after the instruction is decoded. Then, the branch instruction control unit sets the obtained address mode in a bit for designating the address mode of a branch destination (address mode designation bit).

If the branch instruction is not accompanied by an address mode change, the branch instruction control unit sets the same address mode as that of the branch instruction of a branch source in the address mode designation bit since the address mode of the branch destination is not changed.

Alternatively, an address mode designation bit is not generated in the branch instruction control unit, and instead an address mode designation bit, as well as a branch designation address, can also be generated in the branch destination address generation circuit.

If the instruction processing device is provided with a plurality of instruction fetch ports and an instruction fetch is processed by way of an out-of-order system, a new address mode corresponding to a switched-over port must be designated every time a port is switched over. However, in the conventional

device an address mode was not stored for each port.

In the device of this preferred embodiment, an address mode is configured to be stored for each port. Accordingly, even if the address modes of respective ports are different, a correct instruction process is executed, for example, when an instruction pre-fetch is performed at a specific port and then the instruction fetch is attempted to be restored to a sequential instruction fetch at another port.

Fig. 2 shows the detailed configuration of an instruction processing device of the present invention. An instruction processing device shown in Fig. 2 comprises an instruction address register 11, an instruction address generation circuit 12, an instruction fetch control circuit 13, an instruction cache 14, a branch history 15, an instruction buffer 16, a decoder 17, an instruction control circuit 18, a branch destination address generation circuit 19, a branch instruction control circuit 20 and a PSW control circuit 21.

Out of these, the instruction address register 11, instruction address generation circuit 12, instruction fetch control circuit 13, instruction cache 14 and branch history 15 correspond to an instruction fetch pipeline circuit, and the

instruction buffer 16, decoder 17, instruction control circuit 18, branch destination address generation circuit 19, branch instruction control circuit 20 and PSW control circuit 21 correspond to an instruction execution pipeline circuit.

The instruction fetch pipeline includes an instruction address issuance cycle (IA), a table cycle (IT), a buffer cycle (IB) and a result cycle (IR), and the instruction execution pipeline includes a decoding cycle (D), an address calculation cycle (A), an execution cycle (E) and a writing cycle (W).

The instruction address register 11, instruction address generation circuit 12 and instruction fetch control circuit 13 also form an instruction fetch circuit, which is provided with three instruction fetch ports of A, B and C, respectively. For example, this instruction fetch circuit performs an instruction fetch using port A at the time of operation start, and switches from port A to another port when performing an instruction pre-fetch.

The instruction cache 14 outputs an instruction code corresponding to the instruction address issued from the instruction fetch control circuit 13, to the instruction buffer 16, and the branch history 15 forms pairs of addresses made up of the address of a branch

instruction which has been branched before and the address of a branch destination at that time, and stores the pairs.

The decoder 17 decodes the instruction code stored in the instruction buffer 16 and passes the obtained information to the instruction control circuit 18, branch destination address generation circuit 19 and branch instruction control circuit 20. The instruction control circuit 18 controls instructions other than a branch instruction. The branch instruction control circuit 20 controls a branch instruction, and the PSW control circuit stores a PSW.

Two modes of 31 bits and 24 bits are used as address modes. As shown in Fig. 3, an address mode is handled as a part of an instruction address, and the bit <0> of the instruction address becomes an address mode designation bit.

This address mode designation bit is added to the bit <0> of all instruction addresses as an instruction address mode and is stored together with the instruction address in all cycles. For example, if the address mode designation bit is a logic "1", it indicates a 31-bit mode. If the bit is a logic "0", it indicates a 24-bit mode.

In the description below, out of the bits of an extended instruction address shown in Fig. 3, a portion excluding the address mode designation bit is called an "instruction address" or just an "address". Fig. 4 shows the format of the entry of the branch history 15 in the case where this address mode designation bit is used.

In Fig. 4, a branch instruction address 31 and a branch destination address 32 are related to each other. An address mode designation bit 33 for designating an address mode to be used for a branch instruction process is added to the branch instruction address 31, and an address mode designation bit 34 for designating an address mode to be used for a branch destination instruction process is added to the branch instruction address 32.

Fig. 5 is a flowchart showing the operation of the instruction processing device shown in Fig. 2.

The instruction fetch control circuit 13 provides the branch history 15 with a generated instruction address, and the branch history 15 searches for a corresponding entry using the given instruction address as a key (step S1).

If the instruction address does not hit the branch history 15, the pre-fetch of a branch

destination instruction is not performed. If in the branch instruction control circuit 20, it is found that the branch instruction is branched (step S2), the branch instruction control circuit 20 receives both a branch destination address and a branch destination address mode from the branch destination address generation circuit 19 and generates a branch destination address mode designation bit (step S3). Then, the branch instruction control circuit 20 transmits both the branch destination address and branch destination address mode designation bit to the instruction fetch control circuit 13.

The instruction fetch control circuit 13 performs the instruction fetch of a branch destination based on the branch destination address and branch destination address mode received from the branch instruction control circuit 20 (step S4).

If the instruction address hits the branch history 15, the instruction pre-fetch of a branch destination is performed at another port. If in the branch instruction control circuit 20, it is found that the branch instruction is branched (step S5), the branch instruction control circuit 20 receives both a branch destination address and a branch destination address mode from the branch destination address

generation circuit 19 and generates a branch destination address mode designation bit (step S6).

Then, the branch instruction control circuit 20 judges whether the branch destination address predicted by the branch history 15 and branch destination address mode designation bit are correct (step S7). If they are correct, the branch instruction control circuit 20 considers the prediction to have succeeded and continues to process the branch destination instruction (step S8). If they are not correct, the branch instruction control circuit 20 considers the prediction to have failed and instructs the instruction fetch control circuit 13 to perform a re-fetch (step S9).

If in step S5, it is found that the branch instruction is not branched, the branch instruction control circuit 20 instructs the instruction fetch control circuit 13 to stop the pre-fetch of a branch destination instruction string (step S10). Then, the instruction fetch control circuit 13 restores the fetch port to the original port and continues to process the subsequent instruction string of the branch instruction.

Fig. 6 shows signals used for such an instruction process. The instruction fetch control circuit 13

outputs a signal +IAR<0:31, P0:P3> to the instruction cache 14. The instruction cache 14 outputs a corresponding instruction code to the instruction buffer 16. Out of the signal +IAR<0:31, P0:P3>, +IAR<0>, +IAR<1:31> and +IAR<P0:P3> correspond to the address mode designation bit, instruction address and parity bits of +IAR<0:31>, respectively.

The instruction fetch control circuit 13 also outputs a signal +IAR<0> to the instruction buffer 16, and the instruction buffer 16 outputs the signal to the decoder 17 as +D_IAR<0>.

The instruction buffer 16 also stores a signal indicating the instruction address of the branch instruction +D_BR_IAR<0:31, P0:P3> and outputs this signal to the branch instruction control circuit 20. This signal can be transferred to the branch instruction control circuit 20 via the decoder 17 or it can be directly transferred from the instruction buffer 16 to the branch instruction control circuit 20. If an instruction code outputted from the instruction cache 14 corresponds to a branch instruction, a signal +D_IAR<0> matches a signal +D_BR_IAR<0>.

The decoder 17 outputs a signal +A_ADRS_MODE indicating an address mode designated as one of the

operands of the branch instruction to the branch destination address generation circuit 19 together with a signal +OP_DATA for indicating data used to generate a branch destination address. This signal +A_ADRS_MODE indicates the address mode of a branch destination instruction.

The branch destination address generation circuit 19 generates a signal +EAG2<1:31, P0:P3> indicating the address of a branch destination instruction from the received signal and outputs the signal to the branch instruction control circuit 20 together with a signal +A_BSM_31BIT_ADRS_MODE indicating the address mode of the branch destination instruction. The branch destination address generation circuit 19 also receives a signal +D_BR_IAR<0> indicating the address mode designation bit of the branch instruction from the decoder 17 and, based on the signal, executes a mask process, etc., of the instruction address of the branch destination.

The branch instruction control circuit 20 receives a signal +BRHIS_TIAR<0:31, P0:P3> indicating the address and address mode designation bit of an predicted branch destination instruction from the branch history 15, compares the signal with a signal from the branch destination address generation circuit

19 and judges whether the prediction result is correct.

If the prediction result is judged to be correct, the branch instruction control circuit 20 instructs the instruction fetch control circuit 13 to continue the process of the predicted branch destination. If the prediction result is judged to be wrong, the branch instruction control circuit 20 transmits both a signal +RSBR_TIAR<0:31, P0:P3> indicating the address and address mode designation bit of the correct branch destination instruction and an instruction re-fetch request to the instruction fetch control circuit 13.

Next, the detailed configuration and operation of the instruction processing device shown in Fig. 2 are described with reference to Figs. 7 through 12.

When the process is re-started after the first instruction fetch and an interruption, the instruction fetch control circuit 13 acquires the address mode information from the PSW control circuit 21 and sets the information in the address mode designation bit of the instruction fetch circuit. Thus, an address mode is set in each of the instruction address register 11, instruction address generation circuit 12 and instruction fetch control circuit 13.

Fig. 7 shows a circuit for storing an address mode designation bit in the instruction address generation circuit 12. Latch circuits 41, 42 and 43 shown in Fig. 7 store the address mode designation bits and instruction addresses to be fetched of ports A, B and C, respectively. The input signal +IF_EAG<0:31> of each latch circuit indicates both a generated address mode designation bit and an instruction address, and the output signals +IARA<0:31>, +IARB<0:31> and +IARC<0:31> correspond to the signal +IAR<0:31> shown in Fig. 6.

If an instruction fetch port A is to be used at the time of start, the address mode used when an instruction fetch request is issued, is inputted periodically and is stored until the clear request of the port is issued. The address mode is updated every time an instruction fetch request is issued. Therefore, if the address mode is changed on the way, the changed address mode is stored.

The operation performed when an instruction fetch port is switched over is described next. Here, as one example, the switchover of an instruction fetch port to perform an instruction pre-fetch accompanied by a branch prediction is described.

When the branch prediction of a branch

instruction is performed, the instruction fetch control circuit 13 searches for a fetched instruction address in the branch history 15 and checks whether an instruction with the address has ever been branched before. If the instruction is branched, the instruction fetch control circuit 13 determines the address in which the instruction is branched and issues an instruction fetch request of the instruction address of a branch destination.

In this case, the address mode as well as instruction address of the branch destination must be predicted. In this preferred embodiment, as shown in Figs. 3 and 4, both the instruction address and address mode of the branch destination can be simultaneously predicted by handling the address mode as a part of the address.

Generally speaking, both a branch prediction utilizing the branch history 15 and an instruction fetch of the predicted branch destination are performed prior to an instruction decoding. Therefore, it is unknown whether the predicted instruction is really a branch instruction, whether the branch is actually taken even if it is a branch instruction and whether the instruction address of the predicted branch destination is correct. Therefore, the

instruction fetch of the predicted branch destination must be performed using an instruction fetch port different from the port presently being used.

If an instruction is not branched, and subsequent instruction string is sequentially executed, and specifically if a sequential instruction fetch is performed, ports shall not be switched over.

A case where a branch prediction is made, where an instruction pre-fetch is performed, where it is found that an instruction is branched after an instruction is decoded, and further, where the instruction address of a predicted branch destination is correct, is considered first. This corresponds to the case where the operation in step S8 shown in Fig. 5 is performed.

In this case, an instruction fetch after the branch is performed using a switched-over port (in this example, port B). The instruction pre-fetch of a branch destination based on the branch prediction is performed at a port different from that used for a sequential instruction fetch. In this example, ports A and B are used for a sequential instruction fetch and the instruction pre-fetch of a branch destination, respectively.

At this time, the instruction fetch control

circuit 13 stores the respective address mode designation bits at ports A and B. When it is found that the branch instruction is branched after the branch instruction is decoded, and it is found that the instruction address of a predicted branch destination is correct, port B, instead of port A, becomes a sequential instruction fetch port. At this point, it becomes unnecessary for the instruction fetch circuit to store the address mode of port A.

A case where a branch prediction is made, where the pre-fetch of an instruction is performed, where it is found that an instruction is branched after the instruction is decoded, and further, where the instruction address of a predicted branch destination is wrong, is considered next. This corresponds to the case where the operation in step S9 shown in Fig. 5 is performed. In this case, operation up to the point of the instruction pre-fetch is the same as that described above.

If it is found that an instruction is branched after the instruction is decoded and it is found that the instruction address is wrong, a branch destination instruction fetch must be performed again. At this time, the instruction fetch control circuit 13 performs a instruction re-fetch using both the branch

destination address generated by the branch destination address generation circuit 19 and the address mode designation bit of the branch destination generated by the branch instruction control circuit 20. Thus, all the ports of the instruction fetch control circuit are cleared and an instruction fetch is performed.

A case where a branch prediction is made, where the pre-fetch of an instruction is performed, and where it is found that the instruction is not branched after the instruction is decoded, is considered next. This corresponds to the case where the operation in step S10 shown in Fig. 5 is performed. In this case, operation up to the point of the instruction pre-fetch is the same as that described above.

If a predicted instruction is not branched after the instruction is decoded, and specifically if a predicted instruction is not a branch instruction, or if a branch condition is not met although a predicted instruction is a branch instruction, the sequential instruction fetch is re-started at the original port where the sequential instruction fetch was performed (in this example, port A).

In this case, the address mode designation bit

stored at the port A of the instruction fetch control circuit 13 is used as an address mode designation bit to be used at an instruction fetch request. The port at which an instruction pre-fetch was performed based on the branch prediction (in this example, port B) is cleared.

If an instruction fetch port used for an instruction pre-fetch accompanied by a branch prediction is switched over, an address mode designation bit is set by the operation described above. Next, an address mode designation bit generation circuit for a branch destination is described.

If a branch instruction is accompanied by an address mode change, a signal for an address mode designated by the branch instruction is generated by a circuit, as shown in Fig. 8. In Fig. 8, a signal for an instruction BSM is shown, and this circuit is provided in the branch instruction address generation circuit 19.

In Fig. 8, an input signal +A_BSM indicates an instruction to change an address mode if it is a logic "1", and indicates an instruction not to change the address mode if it is a logic "0". An input signal +BR2_BIT32 indicates a 31-bit mode if it is a logic

"1" and indicates a 24-bit mode if it is a logic "0".

An input signal -BR2_BIT32 corresponds to the negation of a signal +BR2_BRT32. If the signal is a logic "1", it indicates a 24-bit mode, and if the signal is a logic "0", it indicates a 31-bit mode. These signals are generated from the signal +A_ADRS_MODE shown in Fig. 6 in the branch instruction address generation circuit 19.

An AND circuit 51 outputs the logical product of the signals +A_BSM and -BR2_BIT32 as a signal +A_BSM_24BIT_ADRS_MODE. Therefore, the output signal +A_BSM_24BIT_ADRS_MODE indicates that an address mode is changed to a 24-bit mode if it is a logic "1".

An AND circuit 52 outputs the logical product of the signals +A_BSM and +BR2_BIT32 as a signal +A_BSM_31BIT_ADRS_MODE. Therefore, the output signal +A_BSM_31BIT_ADRS_MODE indicates that the address mode is changed to a 31-bit mode if it is a logic "1".

However, if there are only two address modes of a 24-bit mode and 31-bit mode, the instruction process can be controlled using only either a signal +A_BSM_24BIT_ADRS_MODE or a signal +A_BSM_31BIT_ADRS_MODE. Thus, in Fig. 6, only a signal +A_BSM_31BIT_ADRS_MODE is outputted to the branch instruction control circuit 20.

A signal of an address mode of a branch instruction itself is generated by a circuit, as shown in Fig. 9, in the branch instruction control circuit 20. A latch circuit 61 shown in Fig. 9 latches the signal +D_BR_IAR<0> shown in Fig. 6 and outputs the signal as a signal +RSBR_31BIT_ADRS_MODE. The signal +RSBR_31BIT_ADRS_MODE indicates that the address mode of a branch instruction is a 31-bit mode if it is a logic "1".

Fig. 10 shows a circuit for generating the address mode designation bit of a branch destination. This circuit is provided in the branch instruction control circuit 20 and generates an address mode designation bit depending on whether a branch instruction to be executed is accompanied by an address mode change.

If the branch instruction is accompanied by an address mode change, the circuit sets a signal of an address mode designated by the branch instruction, which is received from the branch destination address generation circuit 19, as an address mode designation bit. If the branch instruction is not accompanied by an address mode change, the circuit sets the address mode of the branch instruction as it is, as an address mode designation bit.

In Fig. 10, an input signal +RSBR_BSM_VAL indicates that a decoded branch instruction is accompanied by an address mode change if it is a logic "1". An input signal -RSBR_BSM_VAL corresponds to the negation of the signal +RSBR_BSM_VAL and indicates that the decoded branch instruction is not accompanied by an address mode change if it is a logic "1". These input signals are generated from the decoding result of the decoder 17.

An AND circuit 71 outputs the logical product of the signals +RSBR_BSM_VAL and +A_BSM_31BIT_ADRS_MODE shown in Fig. 8 to an OR circuit 73. An AND circuit 72 outputs the logical product of the signal -RSBR_BSM_VAL and the signal +RSBR_31BIT_ADRS_MODE shown in Fig. 9 to an OR circuit 73. Then, the OR circuit 73 outputs the logical sum of the outputs of these two AND circuits as a signal +EAG2<0> indicating the address mode designation bit of a branch destination.

Therefore, the output signal +EAG2<0> indicates that the address mode of a branch destination is a 31-bit mode if it is a logic "1" and indicates that the address mode of a branch destination is a 24-bit mode if it is a logic "0".

This signal +EAG2<0> is added to the signal

+EAG2<1:31> indicating a branch destination address shown in Fig. 6, and a signal +EAG2<0:31, P0:P3> is generated. The signal +EAG2<0:31, P0:P3> indicates both the address mode designation bit and the branch destination address of a branch destination and is stored in the branch instruction control circuit 20.

Next, a circuit for judging whether both the instruction address and address mode designation bit of a predicted branch destination are correct, is described.

Fig. 11 shows such a judgment circuit, which is provided in the branch instruction control circuit 20. In Fig. 11, an input signal +BRHIS_TIAR<0:31, P0:P3> indicates both a branch destination address and a branch destination address mode designation bit predicted by a branch prediction.

An EXNOR circuit 81 compares the signal +EAG2<0:31,P0:P3> with the signal +BRHIS_TIAR<0:31, P0:P3> bit by bit and outputs a signal indicating the negation of the exclusive logical sum of two corresponding bits. In this preferred embodiment, since an address mode designation bit is handled as a part of an address, an address mode designation bit, a branch destination address and parity bits are simultaneously compared.

An AND circuit 82 outputs the logical product of the comparison results (<0>, <1>, ..., <P3>) of all bits outputted by the EXNOR circuit 81 as a signal +RSBR_TIAR_MATCH.

Therefore, if all of the address mode designation bit, branch destination address and parity bits match, the signal +RSBR_TIAR_MATCH becomes a logic "1". In this case, the branch instruction control circuit 20 judges that the prediction has succeeded and continues the process without modification.

However, if one of those bits does not match, this signal becomes a logic "0" and it is judged that the branch prediction has failed. In this case, the branch instruction control circuit 20 outputs both a branch destination address generated by the branch destination address generation circuit 19 and an address mode designation bit generated by the branch instruction control circuit 20 to the instruction fetch control circuit 13 as the signal +RSBR_TIAR<0:31, P0:P3> shown in Fig. 6. Thus, the instruction fetch of the branch destination is re-started.

When finishing the control of a branch instruction, the branch instruction control circuit 20 also judges whether the branch instruction is

accompanied by an address mode change and outputs the judgment result to the PSW control circuit 21. Thus, since information about a branch instruction accompanied by an address mode change is stored in the PSW control circuit 21, the instruction processing device can be controlled while the instruction by which the address mode is being changed is determined.

When the instruction re-fetch request of a branch destination is issued, the branch instruction control circuit 20 also outputs the address mode designation bit as well as the instruction address of the branch destination to the PSW control circuit 21. Thus, since the address mode as well as the instruction address of the branch destination are stored in the PSW control circuit 21, the instruction processing device can be controlled based on the address mode.

Fig. 12 shows such a circuit for generating an output signal to the PSW control circuit 21. This circuit is provided in the branch instruction control circuit 20. In Fig. 12, an input signal +RSBR_TAKEN indicates that a branch instruction is actually branched if it is a logic "1".

An AND circuit 91 outputs the logical product of the signals +RSBR_TIAR<0:31, P0:P3> and +RSBR_TAKEN as a signal +COMPLETE_RSBR_TIAR<0:31, P0:P3>.

Therefore, this output signal indicates both the address mode designation bit and instruction address of the branch destination existing when the branch instruction is executed and branched.

An AND circuit 92 outputs the logical product of the signal +RSBR_BSM_VAL shown in Fig. 10 and the signal +RSBR_TAKEN as a signal +COMPLETE_RSBR_BSM. Therefore, this output signal indicates that the instruction executed when the branch instruction is branched is a branch instruction BSM which changes the address mode.

These output signals +COMPLETE_RSBR_TIAR<0:31, P0:P3> and +COMPLETE_RSBR_BSM are outputted to the PSW control circuit 21 as control signals.

According to the conventional instruction processing device, the address mode of a branch destination could not be predicted in a branch prediction, and the address mode of a branch destination instruction could not be generated in the calculation of a branch destination address. If the instruction processing device is provided with a plurality of instruction fetch ports, the address mode was not stored for each port. Therefore, if ports are switched over, an instruction which changes the branch destination address mode could not be accurately

controlled by way of hardware. Accordingly, the conventional device required control using a micro-program.

However, according to the instruction processing device of the present invention, address mode information is added to the instruction addresses of all instructions as a part of the instruction addresses, and is stored in all cycles. Accordingly, accurate control is available by way of hardware based on the stored address mode in any cycle.

For example, when the instruction string pre-fetch for the branch destination of a branch instruction accompanied by an address mode change is performed based on a branch prediction, a correct fetch operation can be performed based on the address mode after change. When the instruction address of a branch destination is calculated by an address generator, the address mode of the branch destination can be simultaneously generated.

Furthermore, if the device is provided with a plurality of instruction fetch ports, an address mode is stored for each port. Therefore, even if ports are switched over, a correct fetch operation is available based on the address mode obtained after the switchover.

Since in this way, a branch instruction accompanied by an address mode change can be accurately controlled by way of hardware, there is no need for control using a micro-program and thereby the number of clock cycles needed for an instruction execution can be greatly reduced. Accordingly, the process speed of the instruction processing device can be greatly improved.

In the preferred embodiments described above, two address modes, 24-bit mode and 31-bit mode, are used. However, even if three or more address modes are used, a similar control can be exercised by using a plurality of address mode designation bits.

The present invention can also be applied to an arbitrary mode, which is to be determined at the time of an instruction fetch, of an information processing apparatus other than an address mode. For example, if a branch instruction accompanied by the change of an operation mode, such as an ESA (enterprise system architecture) mode which is specified on the operation specification of a computer, is processed, it is sufficient that information designating an operation mode is added to the instruction addresses of all instructions as a part of the instruction addresses and is stored in all cycles. Thus, control based on

the stored operation mode information becomes available in an arbitrary cycle.

According to the present invention, an instruction accompanied by a mode change can be accurately controlled by way of hardware in an information processing apparatus adopting an out-of-order process system. Accordingly, instruction process speed can be improved.

0360344 0360360

What is claimed is:

1. An instruction processing device, comprising:

a storage circuit storing address mode information of a fetched instruction with an instruction address of the fetched instruction;

a branch instruction control circuit controlling a branch instruction using the address mode information if the fetched instruction is the branch instruction; and

a transfer circuit transferring the address mode information stored in the storage circuit to the branch instruction control circuit when the branch instruction is executed.

2. The instruction processing device according to claim 1, wherein said branch instruction control circuit stores address mode information of a branch destination of the branch instruction with an instruction address of the branch destination.

3. The instruction processing device according to claim 2, wherein said branch instruction control circuit generates the address mode information of the branch destination based on the address mode

information of the branch instruction.

4. The instruction processing device according to claim 2, wherein said branch instruction control circuit judges whether address mode information and an instruction address of a branch destination predicted by a branch prediction are correct using the address mode information and instruction address of the branch destination.

5. The instruction processing device according to claim 2, wherein said branch instruction control circuit outputs a signal indicating the address mode information and instruction address of the branch destination when issuing a branch destination instruction fetch request.

6. The instruction processing device according to claim 1, wherein said branch instruction control circuit outputs a signal indicating whether the branch instruction is accompanied by an address mode change when control of the branch instruction is terminated.

7. The instruction processing device according to claim 1, further comprising

a branch history circuit relating address mode information and an instruction address of a branch instruction to address mode information and an instruction address of a branch destination, storing related address mode information and instruction addresses of the branch instruction and branch destination, and making a branch prediction for the fetched branch instruction.

8. The instruction processing device according to claim 1, further comprising

a branch destination address generation circuit generating an instruction address of a branch destination of the branch instruction using the address mode information,

wherein said transfer circuit transfers the address mode information stored in the storage circuit to the branch destination address generation circuit when the branch instruction is executed.

9. An instruction processing device, comprising:

a storage circuit storing mode information of a fetched instruction with an instruction address of the fetched instruction;

a branch instruction control circuit controlling

a branch instruction using the mode information if the fetched instruction is the branch instruction; and

a transfer circuit transferring the mode information stored in the storage circuit to the branch instruction control circuit when the branch instruction is executed.

10. An instruction processing device, comprising:

a fetch circuit fetching an instruction;

a storage circuit storing mode information of each fetched instruction as a part of an instruction address of the fetched instruction; and

a control circuit controlling an instruction process of each instruction based on the stored mode information.

11. An instruction processing device provided with a plurality of instruction fetch ports and performing an instruction fetch by way of an out-of-order system, comprising:

a storage circuit storing mode information obtained when an instruction fetch request is issued, with an instruction address for each port; and

a fetch circuit performing an instruction fetch based on mode information corresponding to a port to

be used.

12. An instruction processing method, comprising:

handling mode information of an information processing apparatus, which is to be determined when fetching each instruction, as a part of an instruction address;

fetching an instruction;

storing mode information of the fetched instruction in each cycle of an instruction process for the fetched instruction; and

controlling the instruction process for the fetched instruction based on the stored mode information.

13. An instruction processing device, comprising:

storage means for storing address mode information of a fetched instruction with an instruction address of the fetched instruction;

branch instruction control means for controlling a branch instruction using the address mode information if the fetched instruction is the branch instruction; and

transfer means for transferring the address mode information stored in the storage means to the branch

instruction control means when the branch instruction is executed.

14. An instruction processing device, comprising:

storage means for storing mode information of a fetched instruction with an instruction address of the fetched instruction;

branch instruction control means for controlling a branch instruction using the mode information if the fetched instruction is the branch instruction; and

transfer means for transferring the mode information stored in the storage means to the branch instruction control means when the branch instruction is executed.

15. An instruction processing device, comprising:

fetch means for fetching an instruction;

storage means for storing mode information of each fetched instruction as a part of an instruction address of the fetched instruction; and

control means for controlling an instruction process of each instruction based on the stored mode information.

16. An instruction processing device provided with a

plurality of instruction fetch ports and performing an instruction fetch by way of an out-of-order system, comprising:

storage means for storing mode information obtained when an instruction fetch request is issued, with an instruction address for each port; and

fetch means for performing an instruction fetch based on mode information corresponding to a port to be used.

Abstract of the Disclosure

In the out-of-order process of an instruction, the address mode information of a fetched branch instruction is automatically transferred from an instruction fetch pipeline to an instruction execution pipeline. If the branch instruction is accompanied by an address mode change, the address mode after change designated by the branch instruction is adopted as the address mode of a branch destination. If the branch instruction is not accompanied by an address mode change, the transferred address mode information is adopted as the address mode of the branch destination.

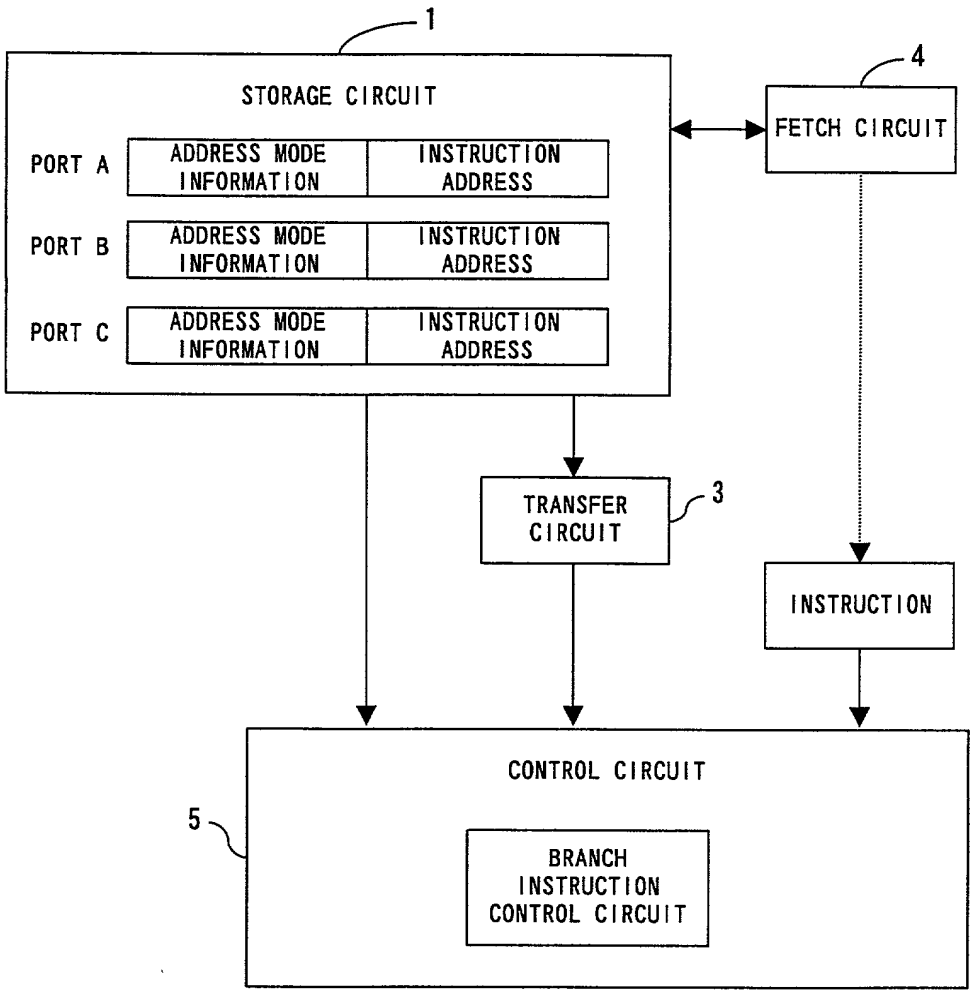
1
STORAGE CIRCUIT
PORT A
ADDRESS MODE INFORMATION
INSTRUCTION ADDRESS
PORT B
ADDRESS MODE INFORMATION
INSTRUCTION ADDRESS
PORT C
ADDRESS MODE INFORMATION
INSTRUCTION ADDRESS
4
FETCH CIRCUIT
TRANSFER CIRCUIT
3
INSTRUCTION
CONTROL CIRCUIT
5
BRANCH INSTRUCTION CONTROL CIRCUIT

FIG. 1

0000000-11202560

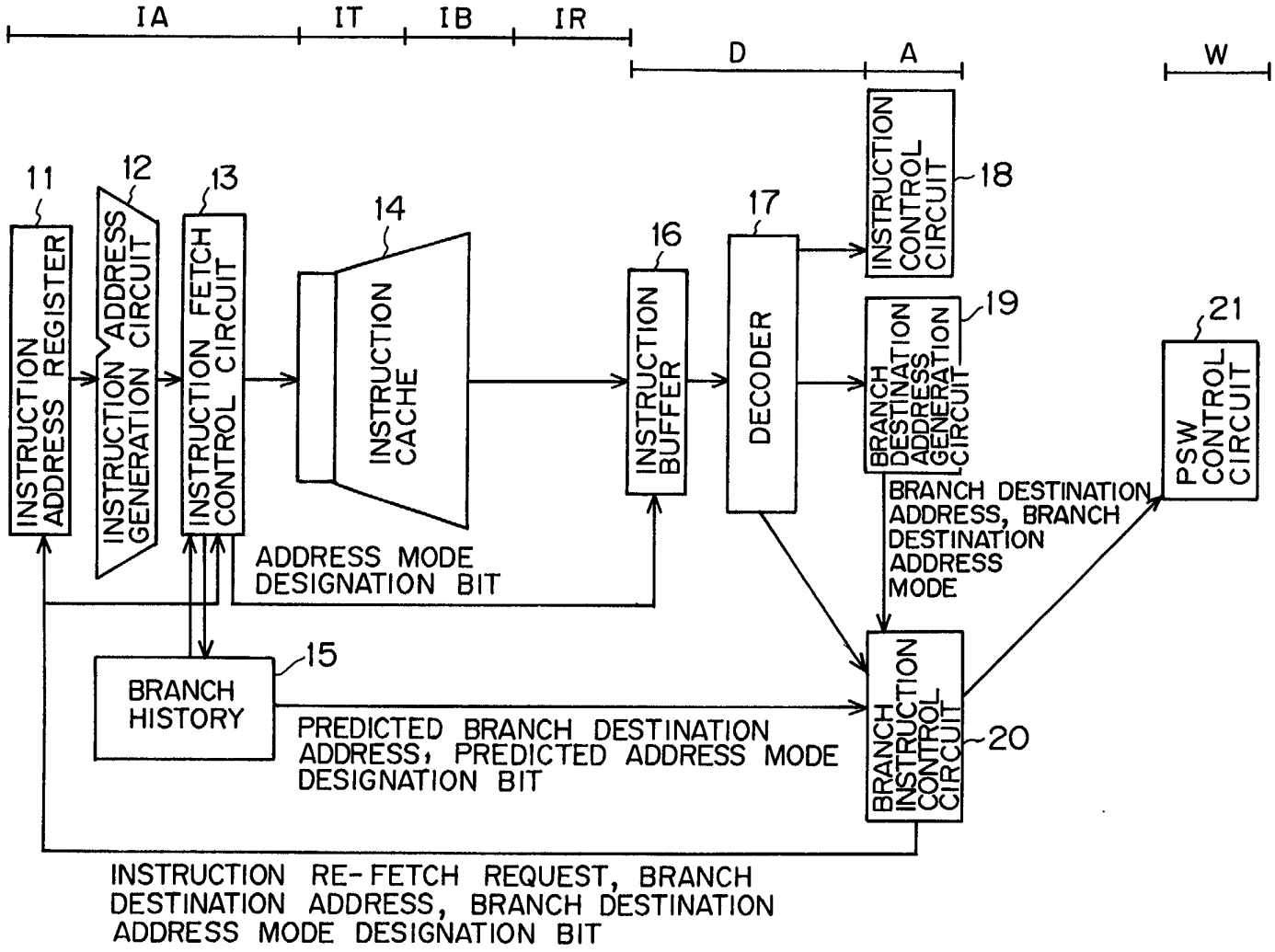

FIG. 2

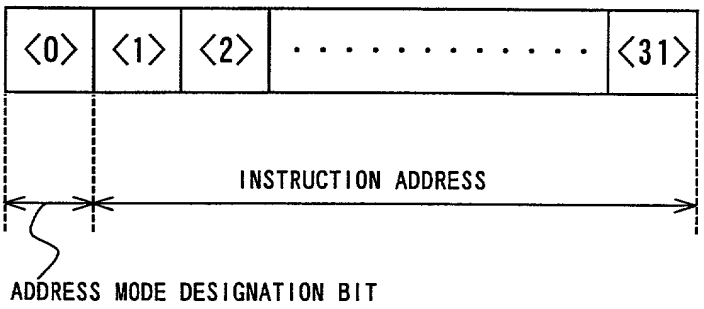

FIG. 3

0030345 030303

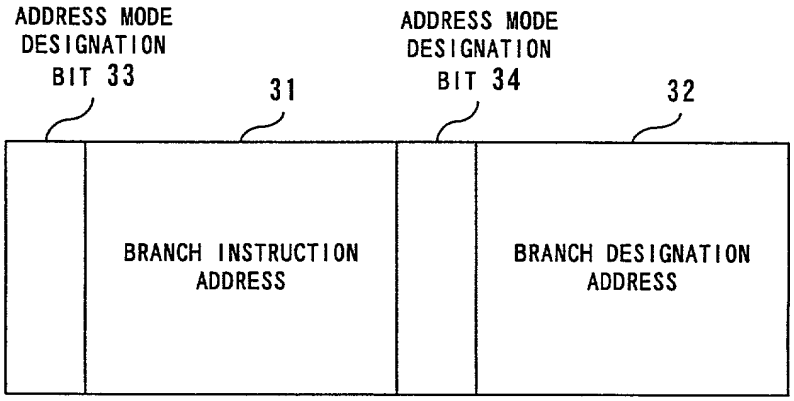

FIG. 4

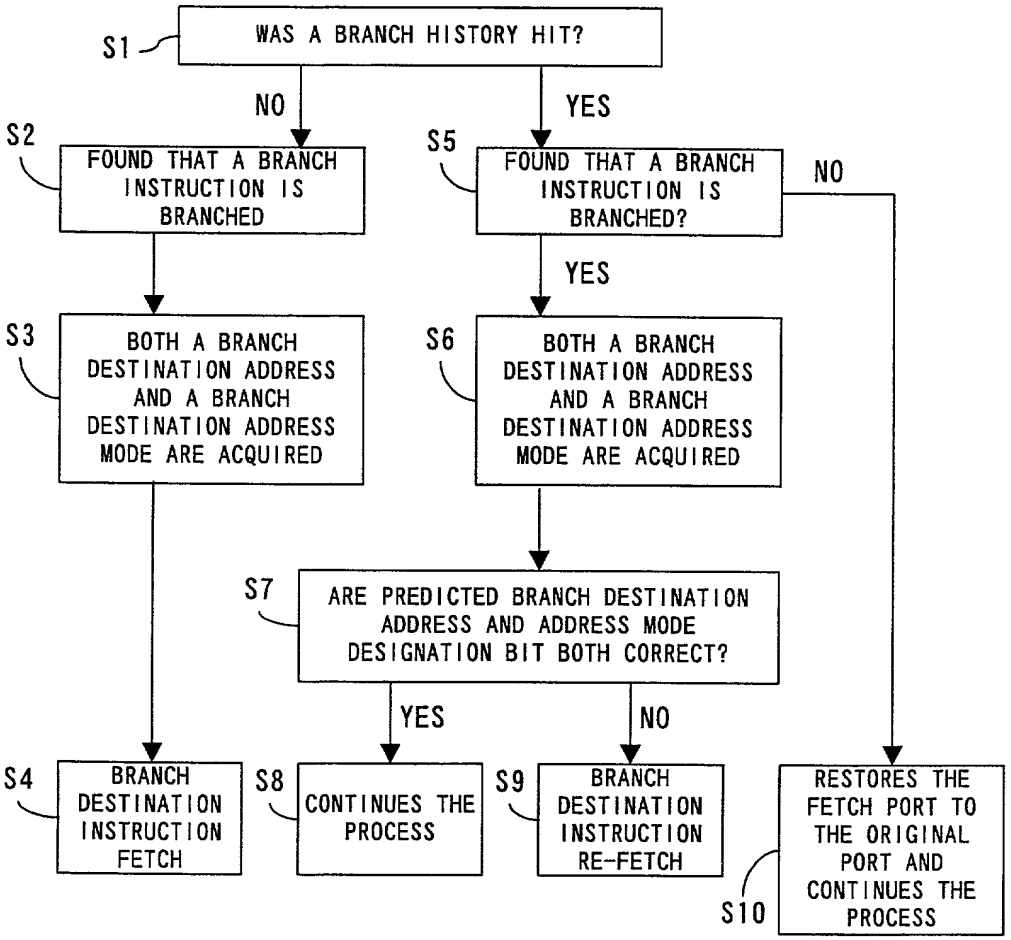
S1
WAS A BRANCH HISTORY HIT?
NO
YES
S2
FOUND THAT A BRANCH INSTRUCTION IS BRANCHED
S5
FOUND THAT A BRANCH INSTRUCTION IS BRANCHED?
NO
YES
S3
BOTH A BRANCH DESTINATION ADDRESS AND A BRANCH DESTINATION ADDRESS MODE ARE ACQUIRED
S6
BOTH A BRANCH DESTINATION ADDRESS AND A BRANCH DESTINATION ADDRESS MODE ARE ACQUIRED
S7
ARE PREDICTED BRANCH DESTINATION ADDRESS AND ADDRESS MODE DESIGNATION BIT BOTH CORRECT?
YES
NO
S4
BRANCH DESTINATION INSTRUCTION FETCH
S8
CONTINUES THE PROCESS
S9
BRANCH DESTINATION INSTRUCTION RE-FETCH
S10
RESTORES THE FETCH PORT TO THE ORIGINAL PORT AND CONTINUES THE PROCESS

FIG. 5

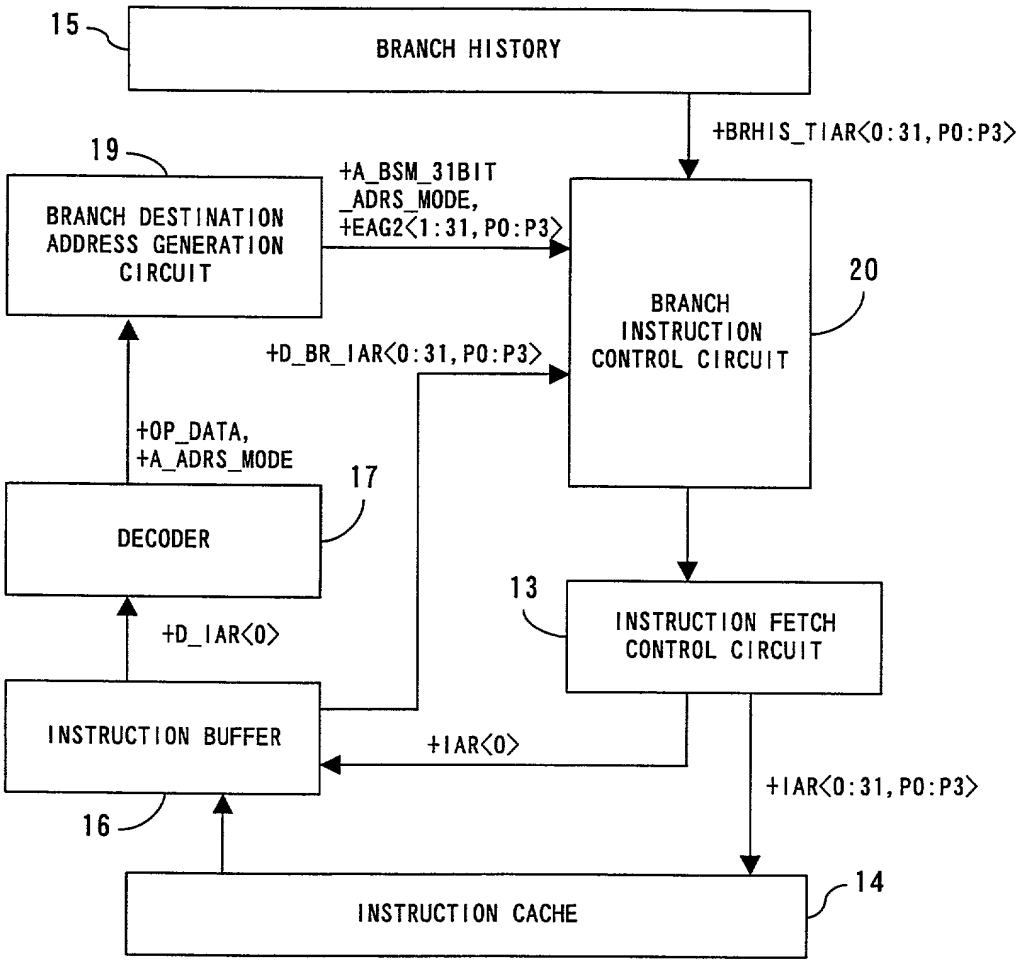
15
BRANCH HISTORY
+BRHIS_TIAR<0:31,P0:P3>
19
BRANCH DESTINATION ADDRESS GENERATION CIRCUIT
+A_BSM_31BIT _ADRS_MODE, +EAG2<1:31,P0:P3>
20
BRANCH INSTRUCTION CONTROL CIRCUIT
+D_BR_IAR<0:31,P0:P3>
+OP_DATA, +A_ADRS_MODE
17
DECODER
13
INSTRUCTION FETCH CONTROL CIRCUIT
+D_IAR<0>
INSTRUCTION BUFFER
+IAR<0>
+IAR<0:31,P0:P3>
16
14
INSTRUCTION CACHE

FIG. 6

0080801T1

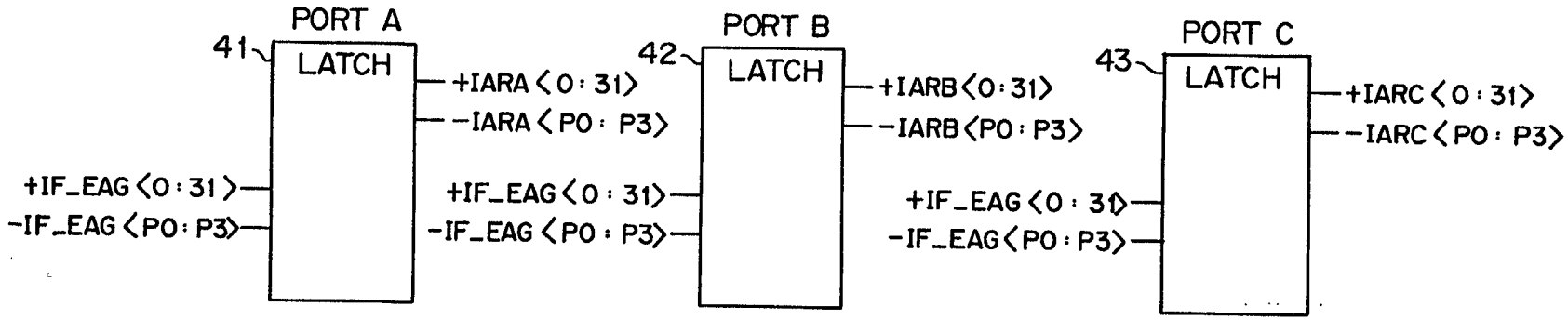

FIG. 7

0030E0 1T20250

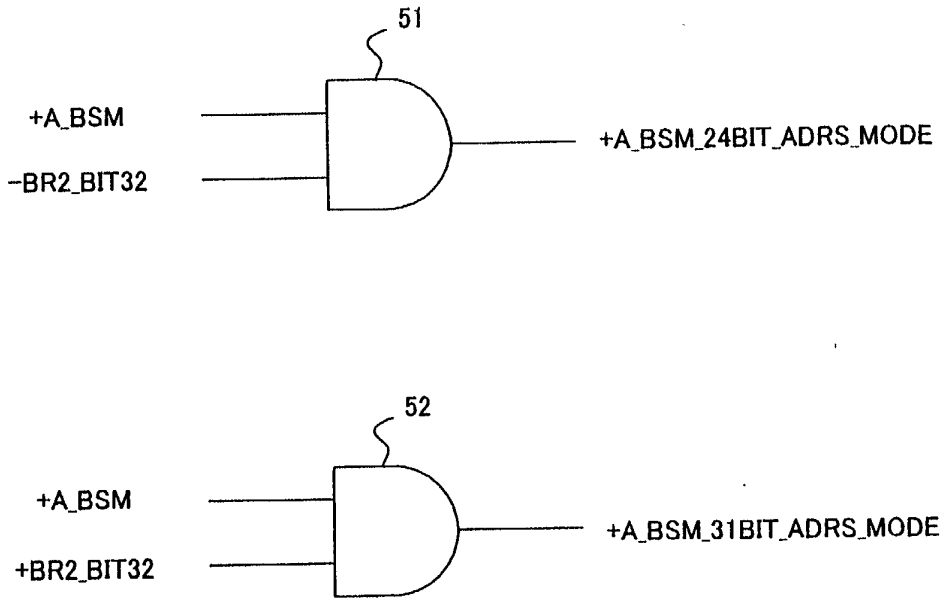

FIG. 8

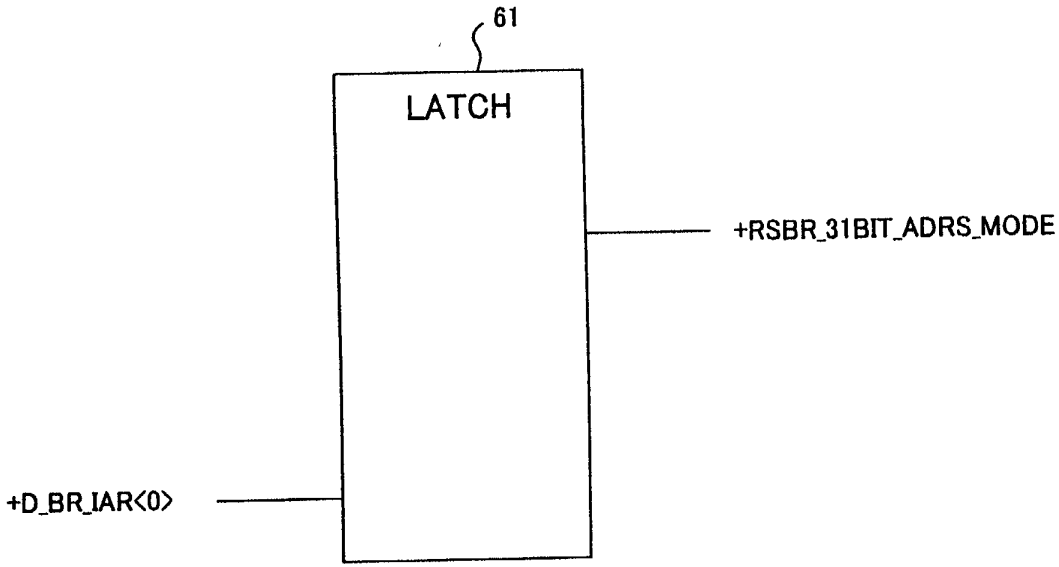

FIG. 9

0002601A1 I

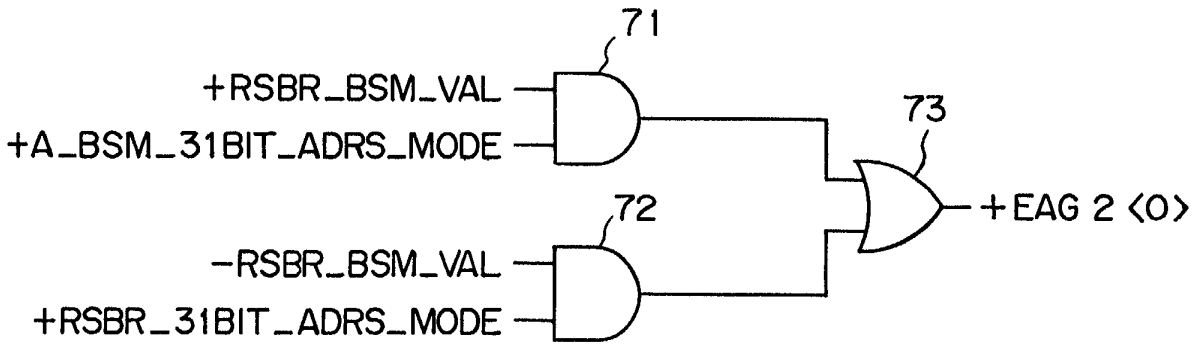

FIG. 10

0020360 A1 20010816

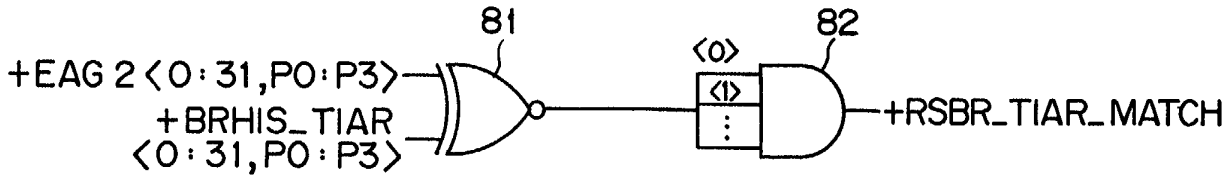

FIG. 11

DOCKET NO. 2000

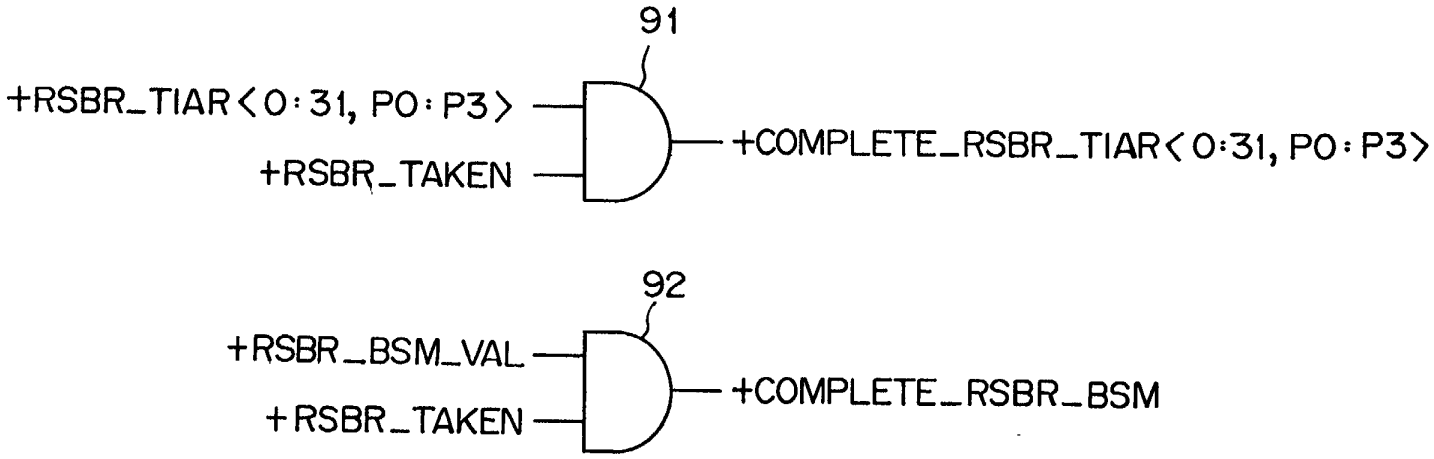

FIG. 12

PTO/SB/106 (6-06)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

Japanese Language Declaration

日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。.

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

INSTRUCTION PROCESSING DEVICE AND METHOD FOR CONTROLLING BRANCH INSTRUCTION ACCOMPANIED BY MODE CHANGE

上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

☐ __月__日に提出され、米国出願番号または特許協定条約国際出願番号を_____とし、
（該当する場合）_____ に訂正されました。

☐ was filed on _____
as United States Application Number or
PCT International Application Number
_____ and was amended on
_____ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第３７編第１条５６項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner of Patents and Trademarks, Washington, DC 20231.

PTO/SB/106 (6-96)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration
(日本語宣言書)

私は、米国法典第３５編１１９条(a)-(d)項又は３６５条(b)項に基と下記の、米国以外の国の少なくとも一ヵ国を指定している特許協力条約３６５(a)項に基ずく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)
外国での先行出願

(Number) (番号)	(Country) (国名)	(Day/Month/Year Filed) (出願年月日)	Priority Not Claimed 優先権主張なし
11-276233	Japan	29th/September/1999	☐
			☐

私は、第３５編米国法典１１９条(e)項に基いて下記の米国特許出願規定に記載された権利をここに主張いたします。

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.) (出願番号)	(Filing Date) (出願日)	(Application No.) (出願番号)	(Filing Date) (出願日)

私は、下記の米国法典第３５編１２０条に基いて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約３６５条(c)に基ずく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第３５編１１２条第１項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第３７編１条５６項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of application.

(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)

私は、私自身の知識に基ずいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じるところに基ずく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第１８編第１００１条に基ずき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに下記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

PTO/SB/108 (8-96)
Approved for use through 9/30/96. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration
(日本語宣言書)

委任状： 私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。（弁護士、または代理人の氏名及び登録番号を明記のこと）

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith *(list name and registration number)*

James D. Halsey, Jr., 22,729; Harry John Staas, 22,010; David M. Pitcher, 25,908; John C. Garvey, 28,607; J. Randall Beckers, 30,358; William F. Herbert, 31,024; Richard A. Gollhofer, 31,106; Mark J. Henry, 36,162; Gene M. Garner II, 34,172; Michael D. Stein, 37,240; Paul I. Kravetz, 35,230; Gerald P. Joyce, III, 37,648; Todd E. Marlette, 35,269; Harlan B. Williams, Jr., 34,756; George N. Stevens, 36,938; Michael C. Soldner, 41,455; Norman L. Ourada, 41,235; Kevin R. Spivak, P-43,148; and William M. Schertler, 35,348 (agent)

書類送付先

Send Correspondence to:

STAAS & HALSEY
700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001

直接電話連絡先： (名前及び電話番号)

Direct Telephone Calls to: *(name and telephone number)*

STAAS & HALSEY
(202) 434-1500

唯一または第一発明者名	Full name of sole or first inventor Ryuichi SUNAYAMA
発明者の署名 日付	Inventor's signature Date Ryuichi Sunayama March 2, 2000
住所	Residence Kanagawa, Japan
国籍	Citizenship Japan
私書箱	Post Office Address c/o FUJITSU LIMITED, 1-1, Kamikodanaka
	4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan
第二共同発明者	Full name of second joint inventor, if any Aiichiro INOUE
第二共同発明者 日付	Second inventor's signature Date Aiichiro Inoue March 2, 2000
住所	Residence Kanagawa, Japan
国籍	Citizenship Japan
私書箱	Post Office Address c/o FUJITSU LIMITED, 1-1, Kamikodanaka
	4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan

(第三以降の共同発明者についても同様に記載し、署名をすること)

(Supply similar information and signature for third and subsequent joint inventors.)